UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,125	02/20/2004	Douglas J. Koslow	CA7037682001	5265

55497 7590 06/23/2008

VISTA IP LAW GROUP LLP
1885 Lundy Avenue
Suite 108
SAN JOSE, CA 95131

EXAMINER	
JACOB, MARY C	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
06/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/784,125	KOSLOW ET AL.	
	Examiner	Art Unit	
	MARY C. JACOB	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 April 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The response filed 4/14/08 has been received and considered. Claims 1-38 have been presented for examination.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 28, 30, 33, 34, 37 and 38 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

4. Claims 28 and 30 recite functional descriptive material and therefore, is directed to non-statutory subject matter. The claims are directed to "a system for simultaneous debugging of an electrical design" and "a system for processing a design that is based upon multiple programming languages" and further recite "means for" language without any hardware elements set forth (for example, a processor or memory). Because the specification, specifically paragraph 0040, last sentence, leads to the conclusion that the claimed "means" appear to cover at least one software embodiment, and the claim does not set forth any hardware elements to show that the claim is directed only to a hardware embodiment, the claim is interpreted to cover a software embodiment. Therefore the claims are interpreted to recite functional descriptive material only, and therefore, are directed to non-statutory subject matter.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-7, 12, 13, 15-17, 27, 28, 31-34 are rejected under 35 U.S.C. 102(b) as being anticipated by Hollander (US Patent 6,182,258).

7. As to Claims 1, 27 and 28, Hollander teaches: a method for debugging an electrical design (column 10, lines 39-43, lines 50-58; column 11, lines 2-5) having both an HDL portion (Figure 5, element 170; column 10, lines 35-36) and a general programming language portion (Figure 5, elements 172, 163; column 10, lines 24-28), comprising: interrupting a simulator that operates upon the HDL portion to allow for debugging of the HDL portion (column 5, lines 44-48; column 9, lines 12-13; column 10, lines 35-36 and lines 50-56), the simulator interrupted by an external debugger (column 5, lines 44-48; column 10, lines 44-50; Figure 6, element 90, "Stop on errors", "Breakpoints"), the external debugger debugging the general language portion (column 10, lines 24-28 and 50-61; column 11, lines 2-16); handling a simulator request with the external debugger for the simulator that is interrupted, the external debugger calling a request processing function at the simulator, the simulator request for simulation of the HDL (column 5, lines 44-48; column 9, lines 58-65; column 10, lines 43-50; column 11, lines 6-13; Figure 6, element 90, "Continue", "Return"); executing the request

processing function at the simulator to respond to the simulator request (column 9, lines 58-65; column 10, lines 47-49); and generating debug results based upon executing the request processing function and storing the debug results in a computer-readable medium (column 9, lines 21-24; column 10, line 65-column 11, line 11; Figure 1, element 24).

8. As to Claims 2, 31 and 33, Hollander teaches: the simulator request accesses a portion of the HDL portion (column 7, lines 27-28; column 13, lines 38-41).

9. As to Claim 3, Hollander teaches: the simulator request accesses HDL signal values (column 9, lines 58-65).

10. As to Claim 4, Hollander teaches: the simulator request accesses HDL design hierarchy (column 7, lines 27-30; column 9, lines 59-65).

11. As to Claim 5, Hollander teaches: the simulator request operates simulator functionality (column 9, lines 59-65; column 10, lines 46-49Figure 6, element 90, "continue", "stop on errors", "breakpoints").

12. As to Claims 6, 32 and 34, Hollander teaches: the general programming language portion comprises C, C++, or SystemC code (Hollander: column 10, lines 27-28).

13. As to Claim 7, Hollander teaches: the HDL portion comprises VHDL or Verilog (Hollander: column 6, lines 37-39 and lines 61-63; column 10, lines 34-35).

14. As to Claim 12, Hollander teaches: the simulator request is generated at a simulator GUI (Hollander: column 10, line 59-column 11, line 8, Figure 6, element 90).

15. As to Claim 13, Hollander teaches: the response to the simulator request is displayed at the simulator GUI (Hollander: column 10, line 59-column 11, line 8, Figure 6).

16. As to Claims 15, Hollander teaches: the simulator request is routed through a debugger GUI for the external debugger (Hollander: column 10, line 59-column 11, line 8, Figure 6).

17. As to Claim 16, Hollander teaches: the simulator request is directly routed to the external debugger (Hollander: column 10, line 59-column 11, line 8, Figure 6).

18. As to Claim 17, Hollander teaches: the request processing function is set up ahead of time at the simulator to handle anticipated simulator requests (Hollander: column 5, lines 44-48; column 11, lines 6-8).

Claim Rejections - 35 USC § 103

19. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each

claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

20. Claims 8-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander as applied to claim 1 above, in view of Chan (US Patent 6,466, 898).

21. Hollander teaches a method for simultaneous debugging of an electrical design wherein a simulator request is handled with an external debugger, the external debugger calling a request processing function at a simulator, the simulator request for simulation of the HDL portion of the design.

22. Hollander does not expressly teach (claim 8) the action of having the external debugger call the request processing function is based upon recognition of a waiting simulator request, (claim 9) recognition of the waiting simulator request is based upon a message sent to the external debugger, (claim 10) recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue, (claim 11) recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue.

23. Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16), wherein the execution of logic simulation events includes wherein (claim 8) the call of a request processing function is based upon recognition of a waiting simulator request (column 7, lines 1-7, lines 32-39; Figure 3, element 14), (claim 9) wherein the recognition of the waiting

simulator request is based upon a message sent to the logic simulation program (column 7, lines 1-7, lines 32-39; Figure 3, element 14), wherein (claim 10) recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue (column 7, lines 1-7, lines 32-46; Figure 3, element 14), and wherein (claim 11) recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue (column 20, lines 55-66; Figure 17, element 89).

24. Hollander and Chan are analogous art since they are both directed to the simulation of a HDL design.

25. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method for simultaneous debugging of an electrical design wherein an external debugger calls a request processing function at a simulator as taught by Hollander to further include wherein the call of a request processing function is based upon recognition of a waiting simulator request, wherein the recognition of the waiting simulator request is based upon a message sent to the logic simulation program, wherein recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue, and wherein recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue as taught in Chan since Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16).

26. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander as applied to claim 1 above, in view of Stallman et al ("Debugging with GDB: The GNU Source-Level Debugger", January 2002, book summary, obtained on www.gnu.org).

27. As to Claim 14, Hollander teaches an external debugger debugging the general programming language portion of a design, the external debugger calling a request processing function at the simulator.

28. Hollander does not expressly teach: the external debugger is a gdb debugger.

29. Stallman et al teaches the gdb, the GNU source level debugger that supports C and C++ among other languages, allows a designer to see what is going on inside a program while it executes or what a program was doing the moment it crashed and allows a designer to catch bugs in a program by: starting the program and specifying anything that might effect the program's behavior, making the program stop under specified conditions, examining what happened when the program stopped and allowing the designer to experiment with changes to see what effect they have on the program (paragraphs 1-3, bullets 1-4).

30. Hollander and Stallman et al are analogous art since they are both directed to the debugging of a general programming language portion of a design.

31. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the external debugger debugging the general programming language portion of the design as taught in Hollander to include the

external debugger being a gdb debugger since Stallman et al teaches that a gdb debugger allows a designer to see what is going on inside a program while it executes or what a program was doing the moment it crashed and allows a designer to catch bugs in a program by: starting the program and specifying anything that might effect the program's behavior, making the program stop under specified conditions, examining what happened when the program stopped and allowing the designer to experiment with changes to see what effect they have on the program (paragraphs 1-3, bullets 1-4).

32. Claims 18-23, 25, 26, 29, 30, 36, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander in view of Chan.

33. As to Claims 18, 29 and 30, Hollander teaches: a method for processing of a design (column 10, lines 39-43, lines 50-58; column 11, lines 2-5) that is based upon multiple programming languages, the multiple programming languages comprising a first language portion (Figure 5, element 170; column 10, lines 35-36) and a second language portion (Figure 5, elements 172, 163; column 10, lines 24-28), in which processing of the second language portion interrupts processing of the first language portion (column 5, lines 44-48; column 10, lines 12-13, lines 44-50), the method comprising: processing the second language portion of the design causing an interruption of processing for the first language portion (column 5, lines 44-48; column 10, lines 12-13, lines 44-50); indicating a need for processing of the second language portion to call a request processing function at the first language portion (column 9, lines 58-65; column 10, lines 44-50); having the processing of the second language portion

call a request processing function at the first language portion that has been interrupted (column 5, lines 44-48; column 9, lines 58-62; column 10, lines 43-50; Figure 6, element 90, "Continue"); executing the request processing function at the first language portion (column 9, lines 58-65; column 10, lines 47-49); and generating processing results based upon executing the request processing function and storing the processing results in a computer-readable medium (column 9, lines 21-24; column 10, line 65-column 11, line 11; Figure 1, element 24).

34. Hollander does not expressly teach: determining whether there are one or more waiting requests for processing of the first language portion.

35. Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16), wherein event-driven logic simulation, as known in the art and by the invention as taught in Chan, includes determining whether there are one or more waiting requests for processing of a portion of the mixed language design (column 7, lines 1-7, lines 32-39; Figure 3, element 14; Figure 8, element 42).

36. Hollander and Chan are analogous art since they are both directed to the simulation of a mixed language design.

37. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the simultaneous processing of a design based on multiple programming languages as taught by Hollander to further include determining whether there are one or more requests waiting for processing a portion of the design

as taught in Chan since Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16) and further teaches that event driven simulation, as known in the art, includes determining whether there are one or more waiting requests for processing of a portion of the mixed language design (column 7, lines 1-7, lines 32-39; Figure 3, element 14; Figure 8, element 42).

38. As to Claim 19, Hollander in view of Chan teach: the one or more waiting requests are for accessing data from the first language portion of the design (Hollander: column 5, lines 44-49; column 9, lines 58-65).

39. As to Claims 20, 36 and 38, Hollander in view of Chan teach: the one or more waiting requests are for debugging the first language portion (Hollander: column 4, lines 45-47; column 5, lines 39-50; column 9, lines 58-65; column 10, lines 43-61).

40. As to Claim 21, Hollander in view of Chan teach: the act of determining whether there are one or more waiting requests for processing of the first language portion is based upon a message sent to a debugger for the processing of the second language portion (Hollander: column 10, lines 24-28 and 50-61; column 11, lines 2-16; Chan: column 7, lines 1-7, lines 32-39; Figure 3, element 14).

41. As to Claim 22, Hollander in view of Chan teach: the act of determining whether there are one or more waiting requests for processing of the first language portion is based a periodic check of a request wait queue for the first language portion (Chan: column 7, lines 1-7, lines 32-46; Figure 3, element 14).

42. As to Claim 23, Hollander in view of Chan teach: the act of determining whether there are one or more waiting requests for processing of the first language portion is based on whether a threshold number of simulator requests are waiting in a request wait queue (Chan: column 20, lines 55-66; Figure 17, element 89).

43. As to Claim 25, Hollander in view of Chan teach: processing the second language portion comprises debugging the second language portion (Hollander: column 10, lines 24-28, lines 39-42, lines 54-56; column 10, line 67-column 11, line 5; Chan: column 10, lines 37-42).

44. As to Claim 26, Hollander in view of Chan teach: the request processing function is set up ahead of time to handle anticipated requests (Hollander: column 5, lines 44-48; column 11, lines 6-8).

45. Claims 24, 35 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander in view of Chan as applied to claim 18 above, further in view of Stallman et al.

46. Hollander in view of Chan teach a method for simultaneous processing of a design that is based upon multiple programming languages that includes handling one or more waiting requests for processing of a first language portion by having the processing of a second language portion call a request processing function at the first language portion that has been interrupted.

47. Hollander in view of Chan does not expressly teach: the request processing function is called by a gdb debugger.

48. Stallman et al teaches the gdb, the GNU source level debugger that supports C and C++ among other languages, allows a designer to see what is going on inside a program while it executes or what a program was doing the moment it crashed and allows a designer to catch bugs in a program by: starting the program and specifying anything that might effect the program's behavior, making the program stop under specified conditions, examining what happened when the program stopped and allowing the designer to experiment with changes to see what effect they have on the program (paragraphs 1-3, bullets 1-4).

49. Hollander in view of Chan and Stallman et al are analogous art since they are directed to the debugging of a general programming language portion of a design.

50. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the handling of one or more waiting requests for processing of the first language portion by having the processing of the second language portion call a request processing function at the first language portion that has been interrupted as taught in Hollander in view of Chan to include the request processing function is called by a gdb debugger since Stallman et al teaches that a gdb debugger allows a designer to see what is going on inside a program while it executes or what a program was doing the moment it crashed and allows a designer to catch bugs in a program by: starting the program and specifying anything that might effect the program's behavior, making the program stop under specified conditions, examining what happened when the program stopped and allowing the designer to experiment with changes to see what effect they have on the program (paragraphs 1-3, bullets 1-4).

Response to Arguments

51. Applicant's arguments filed 4/14/08 have been fully considered but they are not persuasive.

Claim Rejections Under 35 U.S.C. 101

52. Applicants argue that Claims 28 and 30 are not directed to non-statutory subject matter because both claims contain at least one element that is stored on some computer-readable medium (page 8). The Examiner notes that although the claim recite "storing the debug results in a computer readable medium", the "means for" generating and storing the debug results are still interpreted to encompass at least one software embodiment as discussed above. The "computer readable medium" set forth in the claim is not interpreted to be the "means for" the generating and storing.

Claim Rejections Under 35 U.S.C. 102

Claims 1-7, 12-13, 15-17, 27-28 and 31-34

53. Applicant argues that Hollander does not disclose an external debugger debugging a general programming language portion handling a simulator request with the external debugger for the simulator that is interrupted and the external debugger calling a request processing function at the simulator (pages 8-9). The Examiner respectfully disagrees. The cited sections of Hollander describe co-verification wherein an external software program, 168, is run on bus-functional model 172. This external software program is written in a general programming language (*column 10, lines 24-*

28, wherein the bus-functional model on which the external software program is run is written in Verilog, C or is a hardware model, therefore it is understood that the external software program must be written in Verilog or C in order to be run by the bus-functional model; further, as an example, the verification is performed on a driver program which is written in C (column 11, lines 30-32)) and this external software program is debugged (*column 10, lines 52-56, "debugging...all combine information from both the hardware and software sides..."; column 11, lines 2-5, "By debugging…the user can see what happens on the external program side…"; column 11, line 11, "...the user can debug the external software..."*) by an external debugger (*Figure 5, element 166 wherein the apparatus is connected to the co-verification extension module through an interface, 180, and therefore is external to the general language portion*). Further, the external debugger calls a request processing function at the simulator and handles a simulator request for the simulator that is interrupted (*specifically, column 5, lines 44-48, "…The user can set breakpoints…and can observe and change variable values inside...the HDL code"; column 9, lines 58-65, "...The invention can drive or sample any simulator signals...Verilog tasks or VHDL procedures can also be called..."; column 10, lines 35-36, "There herein described apparatus 166 interfaces with a simulator, such as a Verilog model 170, of the DUT"; column 10, lines 43-50, "…A co-verification request is then sent through the socket to the invention. The invention interprets the co-verification request and executes the appropriate functions"; column 11, lines 6-13 and Figure 6, element 90*) wherein it is understood by the Examiner that calling Verilog tasks or VHDL procedures by the invention is calling a request processing function at the simulator,

and that the "breakpoints" will cause an interrupt of the simulator so that variable values within the HDL can be observed. Therefore, it is the Examiner's conclusion that the claim limitations are anticipated by Hollander.

Claim Rejections Under 35 U.S.C. 102

Claims 8-11

54. Applicants argue that Hollander in view of Chan fails to disclose, teach or suggest all of the limitations of Claim 1, from which Claims 8-11 depend (page 9). As discussed above in reference to Claim 1, it the Examiner's conclusion that Hollander in view of Chan teach or suggest all of the limitations set forth by the claims.

Claim 14

55. Applicants argue that Hollander in view of Stallman fail to disclose, teach or suggest all the limitations of claim 1, from which claim 14 depends (page 10). As discussed above in reference to Claim 1, it the Examiner's conclusion that Hollander in view of Chan teach or suggest all of the limitations set forth by the claims.

Claims 18-23, 25-26, 29-30, 36 and 38

56. Applicants argue that Chan does not teach determining whether there are one or more waiting requests for processing of the first language portion and indicating a need for processing of the second language portion to call a request processing function at the first language portion (page 11). The Examiner sets forth that the teachings of

Hollander, not Chan, set forth "indicating a need for processing of the second language portion to call a request processing function at the first language portion" (specifically, column 9, lines 58-65; column 10, lines 44-50) wherein Hollander teaches that the processing of the second language portion (external program, 163) "...reaches pre-designated points at which the program interacts with the DUT...", or the first language portion (170). The Examiner understands that when the pre-designated points are reached, this "indicates" the need for the second language portion to interact or "call a request processing function at the first language portion". This occurs through the invention (Figure 4, 162, Figure 5, 166), wherein the invention can call Verilog procedures or VHDL tasks ("request processing function"). Further, Chan sets forth determining whether there are one or more waiting requests for processing a simulation (column 7, lines 1-7, lines 32-39; Figure 3, element 14; Figure 8, element 42). Therefore, the Examiner concludes that the teachings of Hollander as modified by Chan teach or suggest all of the claim limitations of claims 18, 29 and 30.

Conclusion

57. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

58. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary C. Jacob whose telephone number is 571-272-6249. The examiner can normally be reached Tuesday-Thursday, 7AM-4PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Mary C Jacob/
Examiner, Art Unit 2123

/M. C. J./
6/17/08

/Paul L Rodriguez/
Supervisory Patent Examiner,
Art Unit 2123